

CLAIMS

1. Method for decoding a convolutionally coded input data signal y comprising

- multiplying the input data signal with a scaling factor L_c ;
- 5 - demultiplexing the multiplied input data signal $L_c y$;
- turbo decoding the demultiplexed input data signal $L_c S$ in order to obtain decoder output likelihood ratio data,

characterized in that, the scaling factor L_c is updated for a next iteration in dependence on a combination of a posteriori likelihood data based on turbo decoded output data Λ

10 and a priori likelihood data based on the demultiplexed signal $L_c S$.

2. Method according to claim 1, in which the scaling factor L_c is updated using an estimate of the mean value of the signal amplitude \hat{c} and an estimate of the noise variation $\hat{\sigma}_{n'}^2$.

15 3. Method according to claim 2, in which the scaling factor is updated according to $\hat{L}_c = \frac{2}{\hat{c} \cdot \hat{\sigma}_{n'}^2} \cdot L_c$, in which \hat{L}_c is the updated scaling factor.

4. Method according to claim 2 or 3, in which the estimate of the mean value of

20 the signal amplitude is equal to $\hat{c} = \frac{1}{N} \sum_{i=0}^{N-1} \text{sgn}(\Lambda_i) \cdot L_c \cdot s_i$,

where N is the number of bits in a coding block of the input data signal, s_i is the i^{th} systematic bit, \hat{c} is the estimation of the amplitude of the scaled systematic bits $L_c \cdot s_i$ and Λ_i is the log-likelihood ratio resulting from the most recent turbo decoder iteration.

25 5. Method according to claim 2, 3 or 4, in which the noise variance estimation $\hat{\sigma}_{n'}^2$ equals $\hat{\sigma}_{n'}^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (s'_i - 1)^2 \cdot P_i(1) + (s'_i + 1)^2 \cdot P_i(0) - K$;

the probability of the i^{th} bit being zero is estimated like $\Pr\{x_i = 0\} = P_i(0) = \frac{1}{1 + e^{-\Lambda_i}}$

and the probability of that bit being one like $\Pr\{x_i = 1\} = P_i(1) = \frac{1}{1 + e^{\Lambda_i}} = 1 - P_i(0)$;

the normalised systematic bits s'_i are calculated as $s'_i = \frac{L_c \cdot s_i}{\hat{c}}$;

and where K is a bias correction computed as $K = \frac{1}{N} \sum_{i=0}^{N-1} 2 \cdot (P_i(0) - P_i(1)) \cdot s'_i - 2$.

6. Method according to one of the proceeding claims, in which the scaling factor
10 L_c is initialized either as a fixed value, the result of an initial number of iterations using
a known algorithm, filtering over subsequent iterations and coding blocks, or SNR/SIR
estimation at the input data signal y .

7. Method according to one of the proceeding claims, further comprising
15 - calculating the variation of the scaling factor in subsequent iterations and, when the
variation after a predetermined number of iterations is above a predetermined
threshold value, reverting to a different scaling factor calculation method and/or
turbo decoding method.

20 8. Decoder device for decoding a convolutionally coded input data signal
 y comprising

- a multiplication element (8) for multiplying a received input data signal y with a scaling factor L_c ;
- a demultiplexer (6) for demultiplexing the multiplied input data signal $L_c y$;

25 - a turbo decoder (5) for decoding the demultiplexed input data signal in order to obtain decoder output likelihood ratio data Λ , characterized in that, the decoder device (10) further comprises an adaptive scaling element (7) which is arranged to update the scaling factor L_c for a next iteration based on a combination of a posteriori likelihood data based on turbo decoded output data Λ and a priori likelihood data based on the demultiplexed signal $L_c S$.

9. Decoder device according to claim 8, in which the adaptive scaling element (7) is further arranged to execute the method according to one of the claims 2 through 7.

5 10. Computer program product, which comprises computer executable code, which when loaded on a processing system, provides the processing system with the capability to execute the method according to one of the claims 1 through 7.